PATENT APPLICATION
ATTORNEY DOCKET NO. 03226.110001;P6194

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: **FINDING WORST CASE AGGRESSOR SET FOR LOGICALLY EXCLUSIVE AGGRESSORS DURING NOISE ANALYSIS**

APPLICANT: **Devendra VIDHANI**
Joseph C. FERGUSON

22511
PATENT TRADEMARK OFFICE

"EXPRESS MAIL" Mailing Label Number: EL656799733US
Date of Deposit: June 6, 2001

FINDING WORST CASE AGGRESSOR SET FOR LOGICALLY EXCLUSIVE AGGRESSORS DURING NOISE ANALYSIS

Background of Invention

[0001] For large, high performance processor designs, noise failures are a significant design and verification concern. Due to non-uniform scaling of interconnects, cross-coupling capacitance between wires results in a considerable fraction of total wire capacitance, causing an increase in cross-coupled noise effects, i.e., "crosstalk." At the same time, the quest for higher performance circuits pushes designers to use more aggressive but less noise immune circuit structures, such as dynamic logic and unbuffered latches. The combination of high cross-coupling noise and noise sensitive circuit structures results in a significant noise problem, making effective noise analysis methods critical.

[0002] During noise analysis, nets, i.e., signals, are divided into two classes: victim nets and aggressor nets. A victim net is a net on which noise is injected by one or more neighboring nets through cross-coupled capacitance. The nets that inject noise onto a victim net are considered to be its aggressor nets. For example, if a first net is in proximity to a second net such that when the value of the second net changes, noise is injected on the first net causing it to glitch, i.e., an electrical spike occurs, then the second net is considered to be the first net's aggressor. Thus, another way to distinguish an aggressor net and a victim net is that the aggressor net is the net that switches state and the victim net is the net that maintains its present state, i.e., is "quiet."

[0003] Note that there is typically a capacitance between a victim net and its aggressor even when the aggressor is not switching. Such a capacitance can be referred to as a "ground capacitance," and although ground capacitances do affect the functionality of a victim net, noise estimation techniques can account for

ground capacitances. However, worst case capacitances created on a victim net due to the switching of its aggressors are more difficult to estimate.

[0004] The undesired behavior induced on a victim net can lead to performance degradation because the noise injected on the victim net often propagates to other parts of the processor causing timing failures and/or circuit malfunction. In order to design around such performance degrading effects, noise estimation models are implemented to determine an upper bound on the amount of noise that can be induced on one or more victim nets. However, considering that every victim net is potentially an aggressor of another net and that every aggressor net is potentially also a victim net, the amount of computation needed to generate a reasonable estimate of the noise on a processor caused by such victim-aggressor capacitances can be prohibitively expensive.

Summary of Invention

[0005] According to one aspect of the present invention, a method for finding a worst case aggressor set of a victim net based on a plurality of logically exclusive sets comprises forming a first set, using the first set and the plurality of logically exclusive sets to formulate a problem, and solving the problem to determine a worst case aggressor net of the victim net, where the first set comprises an aggressor net of the victim net, and where the worst case aggressor set comprises the worst case aggressor net.

[0006] According to another aspect, a software tool that finds a worst case aggressor set of a victim net comprises a processor, a memory, and software instructions residing in the memory and executable in the processor for performing a series of operations to find a worst case aggressor net based on a plurality of logically exclusive sets.

[0007] According to another aspect, a method for solving a problem to find a worst

case aggressor net based on a logically exclusive set comprises using a first representation to represent the logically exclusive set, selecting the first representation, selecting a second representation, removing an association of the first representation, removing the first representation, removing an association of the second representation, removing the second representation, and returning the adjacent net represented by the second representation as the worst case aggressor net, where the second representation represents an adjacent net of the first representation.

[0008] According to another aspect, a software tool comprises a processor, a memory, and software instructions residing in the memory and executable in the processor for performing a series of operations for solving a problem to find a worst case aggressor net based on a logically exclusive set.

[0009] According to another aspect, a method for formulating a problem to find a worst case aggressor net of a victim net based on a logically exclusive set comprises using a first representation to represent a net, using a second representation to represent a set, and selectively creating an association between the first representation and the second representation when the net is part of the set, where the net is an aggressor net of the victim net and is part of the logically exclusive set, and where the set is the logically exclusive set.

[0010] According to another aspect, a software tool comprises a processor, a memory, and software instructions residing in the memory and executable in the processor for performing a series of operations for formulating a problem to find a worst case aggressor net of a victim net based on a logically exclusive set.

[0011] Other aspects and advantages of the invention will be apparent from the following description and the appended claims.

Brief Description of Drawings

[0012] Figure 1 shows a flow process in accordance with an embodiment of the present invention.

[0013] Figure 2 shows a graphical model in accordance with the embodiment shown in Figure 1.

[0014] Figure 3 shows a flow process in accordance with the embodiment shown in Figure 2.

[0015] Figure 4a shows an exemplary graphical model in accordance with an embodiment of the present invention.

[0016] Figure 4b shows an exemplary graphical model in accordance with the embodiment shown in Figure 4a.

[0017] Figure 5a shows an exemplary graphical model in accordance with the embodiment shown in Figure 4b.

[0018] Figure 5b shows an exemplary graphical model in accordance with the embodiment shown in Figure 5a.

Detailed Description

[0019] The present invention relates to a method for performing noise analysis based on the logical exclusivity of signals. Further, the present invention relates to a method for performing noise analysis when nets belong to multiple logically exclusive sets. Further, the present invention relates to a method for choosing worst case aggressors from sets of logically exclusive aggressors for noise analysis on a victim net. Further, the present invention relates to a method for finding a worst case aggressor set for logically exclusive aggressors during noise analysis. Further, the present invention relates to a method for basing an estimated amount of noise caused by cross-coupled capacitances in a circuit.

[0020] The present invention uses the idea that for nets in a logically exclusive set, at most only one net can switch states in one direction at a given time. During noise analysis of a "quiet" victim net, if the victim net is capacitively coupled to one or more aggressor nets that are part of a logically exclusive aggressor set, then only one of those aggressor nets can switch at a given time. The coupling capacitances between the victim net and other aggressor nets are considered to be ground capacitances of the victim net because these other aggressor nets are not switching due to the logical exclusivity of the aggressor nets in the aggressor set. Because a processor typically has multiple logically exclusive sets of nets, it becomes necessary to be able to determine worst case aggressors from multiple logically exclusive sets of nets. To this, the present invention provides a method by which to remove from consideration all but the worst case aggressor nets of a particular victim net, where the worst case aggressor nets represent the worst potential case of noise injection on the victim net.

[0021] In order to formulate the problem of determining a worst case aggressor set of a victim net from multiple logically exclusive sets, a graphical model is generated. Referring to Figure 1, an exemplary flow process showing how to generate the graphical model is shown in accordance with an embodiment of the present invention.

[0022] Initially, a victim net having x aggressor nets is chosen for noise analysis (step 10). The x aggressor nets, $a_1, a_2, \ldots, a_x$, belong to a set A (step 12), and have corresponding weights, $w_1, w_2, \ldots, w_x$, which belong to a set W (step 14). Additionally, consider that a processor has m logically exclusive sets of nets, M_1, $M_2, \ldots M_m$, which belong to a set M (step 16).

[0023] Of the x aggressor nets in set A, some may not have logical relationships with other nets, and therefore will not be part of any of the m logically exclusive sets of nets in set M. However, for those aggressor nets in set A that are part of at

least one of the *m* logically exclusive sets of nets in set *M*, a set *N* is created (step 18), where set *N* contains aggressor nets $a_1, a_2, \ldots, a_y$.

[0024] Next, a set *B* is created by subtracting set *N* from set *A* (step 20). Set *B* contains those aggressor nets which are not part of any of the *m* logically exclusive sets of nets in set *M*.

[0025] Thereafter, each of the *m* logically exclusive sets of nets, $M_1, M_2, \ldots M_m$, is respectively reduced to sets $SM_1, SM_2, \ldots SM_m$, such that every net in sets $SM_1, SM_2, \ldots SM_m$ belongs to set *N* (steps 22 and 24), where sets $SM_1, SM_2, \ldots SM_m$ belong to a set *SM*. Those skilled in the art will appreciate that because the reduction of the *m* logically exclusive sets of nets to sets $SM_1, SM_2, \ldots SM_m$ may result in some of the sets in set *SM* containing zero elements, such zero element containing sets can be removed from consideration in the noise analysis of the current victim net (step 26).

[0026] To determine the remaining aggressor nets of the worst case aggressor set, other than the aggressor nets in set *B*, a graphical model, such as a bipartite graph, is generated (step 28). Those skilled in the art will appreciate that other types of graphical models may be used to formulate the problem of determining the worst case aggressors of one or more victim nets. Further, those skilled in the will appreciate that the generating of a graphical model is synonymous with developing, i.e., formulating, a problem that can be methodically solved. Moreover, those skilled in the art will appreciate that the problem associated with the graphical model may actually be implemented by a technique other than a graphical model, e.g., an implementation in software.

[0027] From the graphical model, worst case aggressors are found, where the worst case aggressors found from the graphical model form a set *SV* (step 30). Then, once set *SV* is fully generated, the worst case aggressor set is formed by the union of set *SV* and set *B* (step 32).

[0028] Figure 2 shows an exemplary graphical model (40) in accordance with the embodiment described above with reference to Figure 1. The graphical model, *G* (40), contains nodes for nets $a_1, a_2, \ldots, a_y$, which represent the nets in set *N*, where *y* represents the number of nets in set *N*. Additionally, *G* (40) contains nodes for sets $SM_1, SM_2, \ldots SM_m$, where *m* represents the number of sets in set *SM*. Further, edges between the nodes are created such that there is an edge between a node for a particular net and a node for a particular set if that net belongs to that set. Those skilled in the art will appreciate that although the dotted lines in Figure 2 represent edges, those edges are shown for illustration and may be different based on different net-set relationships. In other words, the edges in Figure 2 were placed arbitrarily in *G* (40) for illustrative purposes, whereas in a graphical model where the net and set values are actually known, edges would be created non-arbitrarily.

[0029] In order to determine the remaining aggressor nets of the worst case aggressor set, other than the aggressor nets in set *B*, a subset of the nets in *G* (40) is determined. Figure 3 shows an exemplary flow process that provides a solution to determining the remaining aggressor nets in accordance with the embodiment shown in Figure 2.

[0030] First, let a set *T* be equal to set *SM* such that set *T* contains sets $SM_1, SM_2, \ldots SM_m$, where *m* represents the number of sets in set *SM* (step 50). Also, initialize an empty set *SV* (step 50), where set *SV* is used to contain the worst case aggressors in *G* (40) and where set *SV* is the subset of the nets in *G* (40). Next, a set from set *T* is chosen (step 52). A determination is then made as to whether the chosen set has any adjacent nets in *G* (40) (step 54), where adjacent nets are those nets to which a set has an edge. If the chosen set does not have any adjacent nets, then that set is removed from set *T* and is also removed from *G* (40) (step 56), where after a determination is made as to whether there are any remaining sets in set *T* (step 64).

[0031] However, if the chosen set has one or more adjacent nets, the adjacent net with the highest weight is chosen (step 58). Thereafter, the chosen set is removed from set *T* and *G* (40) and all edges of the chosen set are removed from *G* (40) (step 60). The chosen net is then added to set *SV* and all edges of the chosen net are removed from *G* (40) (step 62). Next, a determination is made as to whether there are any remaining sets in set *T* (step 64). If there are remaining sets in set *T*, then another set from set *T* is chosen (step 52), and the flow process described above is applied to that set. However, if there are no remaining sets in set *T*, then set *SV* is returned as the set which contains the worst case aggressor nets in G (40) for a particular victim net (step 66).

[0032] An application of the present invention with reference to Figures 1, 2, and 3, is described in the following. Consider that a victim net, v, has aggressor nets a_1, a_2, a_3, a_{15}, and a_{23} which have weights of 15 pF, 20 pF, 10 pF, 5 pF, and 35 pF, respectively, and where a_1, a_2, a_3, a_{15}, and a_{23} form set *A* (step 12) and their corresponding weights form set *W* (step 14). Further, consider that the processor, which v is part of, has logically exclusive sets of nets M_1 and M_2, where $M_1 = \{a_1, a_2, a_4, a_{13}\}$ and $M_2 = \{a_2, a_3, a_9, a_{16}, a_{41}\}$ and where M_1 and M_2 belong to set *M* (step 16). Because a_1, a_2, and a_3 are part of at least one of the logically exclusive sets of nets in set *M*, set *N* is created, where set *N* contains a_1, a_2, and a_3 (step 18).

[0033] Set *B*, which is created by subtracting set *N* from set *A* (step 20), contains a_{15} and a_{23} because these nets are in set *A* but not in set *N*. Next, M_1 and M_2 are respectively reduced to SM_1 and SM_2 such that every net in SM_1 and SM_2 belongs to set *N* (steps 22 and 24). Thus, in this example, $SM_1 = \{a_1, a_2\}$ and $SM_2 = \{a_2, a_3\}$. a_4, a_9, a_{13}, a_{16}, and a_{41} are not included in either SM_1 or SM_2 because none of these nets are part of set *N*. Because neither SM_1 nor SM_2 are empty, they are not removed from consideration (step 26). Thereafter, a graphical problem is formulated (step 28) based on set *SM* and set *N*. A description of this is given with reference to Figures 4a and 4b.

[0034] Those skilled in the art will understand that because SM_1 and SM_2 are logically exclusive, only a_1 and a_3 can switch at a given time or only a_2 can switch at a given time. If a_1 and a_3 switch, then a_2 does not switch, and 25 pF (15 pF from a_1 plus 10 pF from a_3) may be potentially injected on v. If a_2 switches, then a_1 and a_3 do not switch, and 20 pF (20 pF from a_2) may be potentially injected on v. Therefore, the real worst case noise (also referred to as "optimum solution") is 25 pF. Those skilled in the art will appreciate that the present invention provides a method by which to determine a value of potential noise injection that is equal to or slightly greater than the optimum solution.

[0035] Figures 4a and 4b show an exemplary graphical model (70) for the preceding example in accordance with an embodiment of the present invention. Figure 4a shows that the exemplary graphical model (70) is partly created by including nodes for nets a_1, a_2, ..., a_y, which represent the nets in set N, where y represents the number of nets in set N. Figure 4a also shows that the exemplary graphical model (70) is created by including nodes for sets SM_1 and SM_2. For purposes of the description of the exemplary graphical model (70), a node for a net is referred to as a "net" and a node for a set is referred to as a "set."

[0036] Figure 4b shows the edges between the nets and sets of the exemplary graphical model (70) shown in Figure 4a. The edges are created such that there is an edge between a net and a set if that net is a member of that set. For instance, net a_2 belongs to both sets SM_1 and SM_2, and therefore, there is one edge between net a_2 and set SM_1 and another edge between net a_2 and set SM_2.

[0037] Based on the exemplary graphical model (70) shown in Figures 4a and 4b, the flow process described above with reference to Figure 3 is applied. First, set SM_1 is chosen (step 52). Because set SM_1 does have one or more adjacent nets (step 54), the adjacent net with the highest weight is chosen (step 58). Here, net a_2, which has a weight of 20 pF, has the highest weight. Therefore, net a_2 is

chosen.

[0038] Next, set SM_1 and all of its edges are removed from the exemplary graphical model (70) (step 60). Net a_2 is then added to set SV (step 62), and thereafter, net a_2 and all of its edges are removed from the exemplary graphical model (70) (step 62). Figure 5a depicts the exemplary graphical model (70) after set SM_1 and net a_2 and all of their respective edges have been removed.

[0039] After net a_2 and its edges are removed from the exemplary graphical model (70) (step 62), a determination is made as to whether there are any remaining sets in the exemplary graphical model (step 64). In this example, set SM_2 remains, and therefore, set SM_2 is chosen (step 52). Because set SM_2 does have an adjacent net (step 54), net a_3 is chosen (step 58), where net a_3 has the highest weight because it is the only adjacent net of set SM_2.

[0040] Next, set SM_2 and all of its edges are removed from the exemplary graphical model (70) (step 60). Net a_3 is then added to set SV (step 62), and thereafter, net a_3 is removed from the exemplary graphical model (70) (step 62). In the case that net a_3 had any edges, those edges would have been removed from the exemplary graphical model (70). Figure 5b depicts the exemplary graphical model (70) after set SM_2 and net a_3 and all of their respective edges have been removed.

[0041] After net a_3 is removed from the exemplary graphical model (70) (step 62), a determination is made as to whether there are any remaining sets in the exemplary graphical model (70) (step 64). In this example, no sets remain, and therefore set SV is returned (step 30 in Figure 1 and step 66 in Figure 2), where $SV = \{a_2, a_3\}$.

[0042] Based on the values of the weights of nets in set SV, a noise estimate of 30 pF (20 pF from a_2 plus 10 pF from a_3) is made. Those skilled in the art will appreciate that this noise estimate is greater than the optimum solution but less

than a noise estimate which is made by adding the highest p weights of a plurality of aggressor nets, where p represents the number of sets in SM. Thus, the present invention provides a noise estimate that is "pessimistically accurate," i.e., is equal to or slightly greater than the real worst case value.

[0043] The worst case aggressor set for v, which is formed by the union of set SV and set B (step 32), contains a_2, a_3, a_{15}, and a_{23}.

[0044] Advantages of the present invention may include one or more of the following. In some embodiments, because logically exclusive sets of nets are used to determine a worst case aggressor set of a victim net, the potential injection of noise on the victim net can be more accurately analyzed than in cases where logically exclusive sets are not considered.

[0045] In some embodiments, because logically exclusive sets of nets are used in noise analysis, noise on a victim net can be analyzed without removing potential aggressor nets from consideration during noise analysis.

[0046] In some embodiments, because a graphical noise estimation model is used to formulate a problem of determining worst case aggressors of one or more victim nets, more efficient and structured noise analysis can occur relative to non-graphical noise estimation models.

[0047] In some embodiments, because the net with a highest weight is chosen when solving a problem of determining worst case aggressors, the solution is guaranteed to be both pessimistic and accurate, i.e., equal to or slightly greater than a real worst case noise value.

[0048] In some embodiments, solving a formulated problem of determining worst case aggressors for noise analysis is polynomial and may run in real-time.

[0049] While the invention has been described with respect to a limited number of embodiments, those skilled in the art, having benefit of this disclosure, will

appreciate that other embodiments can be devised which do not depart from the scope of the invention as disclosed herein. Accordingly, the scope of the invention should be limited only by the attached claims.